Real-time visio-haptic interaction with static soft tissue models having geometric and material nonlinearity

Igor Peterlik, Mert Sedef, Cagatay Basdogan, Luděk Matyska

Real-time Visio-Haptic Interaction with Static Soft Tissue Models Having Geometric and Material Nonlinearity

Igor Peterlik, Mert Sedef, Cagatay Basdogan, Luděk Matyska

Manuscript received ...

Abstract

Simulation of real-time interactions with soft tissue models is necessary for the development of computer-based medical training systems. To provide realistic visual and haptic feedback to a user by integrating measured material properties into the models, finite element (FE) modeling techniques are typically preferred. However, running a static nonlinear FE model in real-time is a highly challenging task since the resulting stiffness matrix (K) is not constant and varies with the depth of indentation into tissue. We propose a new approach allowing visio-haptic interaction with a FE model of a human liver having both non-linear geometric and material properties. The material properties used in the model are extracted from the experimental data of pig liver to make the simulations more realistic. Our computational approach consists of two main steps: a pre-computation of the configuration space of all possible deformation states of the model, followed by the interpolation of the precomputed data for the calculation of the reaction forces displayed to the user through a haptic device during the real-time interactions. No a priori assumptions or modeling simplifications about the mathematical complexity of the underlying soft tissue model, size and irregularity of the FE mesh are necessary. We show that deformation and force response of the liver in simulations are heavily influenced by the material model, boundary conditions, path of the loading and the type of function used for the interpolation of the pre-computed data.

Index Terms

soft tissue modelling, non-linear model, distributed computations, interpolation methods, haptics

I. INTRODUCTION

Simulating real-time visual interactions with deformable objects in virtual environments is an attractive, but computationally demanding area of research with applications to medical simulation and training. Integrating haptics into these simulations imposes more stringent constraints on the speed and accuracy of computations. While the refresh rate needed for flawless visualization is about 30 Hz, the rate required for haptic rendering of deformable objects is close to 1 kHz due to the higher sensitivity of our haptic channel. Moreover, organ-force models used in medical simulations must be convincing and therefore based on real physics for positive training transfer. However, developing realistic organ-force models is highly challenging not only because of the nonlinearity, rate, and time dependence of an organ's material properties but also because of its layered and non-homogeneous structure. The relations from the theory of elasticity are

usually employed when establishing the mathematical formulation of the problem that is finally solved by some complex computational method such as finite elements (FE). To implement FE method, the geometric model of the organ is divided into surface or volumetric elements, the properties of each element are formulated, and then the elements are combined to compute the organ's deformation states under the influence of external forces such as the ones applied by the surgical instruments. A major advantage of FE modeling is that it uses continuum mechanics and has a solid mathematical foundation. On the basis of the partial differential equations and the constitutive relation used, FE models can accurately approximate static and dynamic deformations of an organ with linear and nonlinear material properties. Another advantage is that FE models require only a few material parameters to describe the realistic response of the organ. However, it is known that simulating "ideal" soft tissue behavior using FE approach within the haptic loop is far beyond the capabilities of nowadays computers. There have been several attempts to address this issue such as the simplification of the underlying mathematical models or employing precomputations before the real-time interaction occurs. Below, we provide a brief summary of the FE approaches used for simulating deformable behaviour of soft objects. The readers can find more extensive coverage of the soft tissue modeling techniques applied to medical simulation in our recent survey paper [1].

A. Related Work

One of the first studies in the field of soft-tissue modeling is reported in [2]. The technique utilizes the FE approach for implementation of a linear model. An update of 30 Hz, equivalent to the visual refresh rate is achieved using the condensation technique. A small-deformation model suitable for laparoscopic surgery is implemented in [3]. In [4] modal analysis is used to reduce the number of computations and achieve real-time visual and haptic update rates. A method based on small area of contact is proposed in [5] for real-time interaction with a deformable static FE model running at haptic refresh rate. While the complexity of on-line computations does not depend on the size of the mesh in this paradigm, it can only be implemented with a linear FE model. Extensive research on soft-tissue modelling is performed within Epidaure project [6]. First, a linear model employing displacement-driven haptic interaction is proposed. It is based on a static superposition of unit displacements which are precomputed in advance. Force extrapolation is used for haptic rendering to compensate for the low update rate of the model [7].

Further, to allow topological changes (such as tearing or suturing), a dynamic mass-tensor model is proposed to simulate linear but anisotropic material. The mass-tensor model is extended by including geometrical non-linearities via Green-StVenant strain tensor in [8], however, linear material properties are assumed and reasonable haptic update rates are achieved by utilizing the force extrapolation technique again. The point-associated finite-field (PAFF) approach presented in [9], also called the finite-spheres method, is a meshless FE approach applied to surgical simulation. PAFF supports simulation of large deformations as well as topology modifications such as cutting, but the brute-force implementation of this technique is still computationally intensive. Geometrically non-linear dynamic model is proposed and implemented in [10]. The model employs mass-lumping for real-time simulation of dynamic behavior, however, only linear material properties are included. Both geometrical and material non-linearities based on Mooney-Rivlin material are modeled in [11]. The implementation is based on dynamic progressive meshes which allow realistic modeling of the deformation in the vicinity of the contact. Although both types of non-linearity are considered, the main limitation of the concept is represented by the initial mesh utilized during the precomputation, which must be dense enough to cover all the expected deformations. Linear viscoelastic model is developed in [12]. The approach is also based on pre-computations. First, the force response of a surface node and the displacements of the neighbouring nodes are recorded for a unit step displacement, being applied for 30 seconds. Second, the recovery of the nodes during 30 seconds is computed and stored after 1 ms unit step force is applied. The two sets of the data are then used for stable haptic interaction. Recently, a new approach based on precomputation is reported in [13]. Although this approach allows the interaction with complex body having non-linear geometry properties, it is implemented with a linear material model (St. Venant) only. The same authors propose a 6DoF haptic rendering of contact in [14]. In this study, the object with complex geometry is considered, nevertheless, the FE model based on reduction is suitable for one type of material.

To summarize, most of the approaches presented above either assume linear material and geometric properties or make modeling assumptions or simplifications to implement nonlinearities with FE models, but then the accuracy of solutions are jeopardized. Integrating both geometric and material nonlinearities into a FE model and simulating realistic visual and haptic interactions with a deformable mesh having complex geometry at real-time rendering rates still remains to be a great challenge.

B. Our Contribution

If linear FE models are used, only small deformations are rendered realistically. However, it is known that soft organ tissues show large deformations under the influence of external forces and the small deformation assumption is not valid for modeling their behavior. In addition to geometric nonlinearities, material properties of soft organ tissues are also nonlinear and our sense of touch is sensitive to these non-linearities as presented in [15]. If modeling assumptions are made to solve nonlinear FE models in real-time, undesired artifacts may appear in the simulations depending on the level of accuracy of these assumptions. We have developed a new precomputation approach allowing haptic interaction with static FE models having both geometric and material non-linearities. The approach is based on a notion of traveling through a configuration space which is precomputed in advance. In the past, we used this approach for the simulation of haptic interactions with biomolecules presented in [16].

The main difference between our approach and the earlier precomputation approaches is that we do not make any assumptions about the mathematical complexity of the underlying soft tissue model, size and irregularity of the FE mesh and the computational complexity of the corresponding solution method. Having a formulation of the boundary problem derived within the theory of elasticity, finite element method is applied in a standard manner resulting in large system of non-linear algebraic equations. These equations must be solved iteratively, which is not possible to do in real-time. In our approach, all these expensive iterative computations are performed in advance during an off-line precomputation phase. This results in a large number of precomputed configurations which are stored to a data file. An advantage of this approach is that any convergence issues which are quite usual in the area of non-linear modeling can be addressed within the off-line calculations. During the real-time interaction, the precomputed configurations are interpolated to calculate the displacements of the nodes and the reaction force vector displayed to the user through a haptic device. The interpolation approach used for the estimation of the force vector is simple enough to execute in the haptic update loop. It is important to emphasize that the approach is not based on the superposition principle, which is not valid for the non-linear models, however, the configurations are “sampled” by standard FE method. Moreover, it is shown, that parameters as location of the interaction point and position of the boundary conditions can significantly influence the force response during the real-time

simulation.

The outline of the paper is following. First, the mathematical background of the problem is briefly presented together with iterative solution method. Further, the integration of the material parameters of the soft tissues into the model is described. In the fourth section, the simulation parameters are identified and the technique based on the precomputation and reconstruction is described.

The fifth section is dedicated to the implementation of the approach. The distributed version of the precomputation is shown and also, the overview of the computational experiments is given. This is followed by evaluation section where the behaviour of the model is briefly studied, the length of the precomputation and accuracy of the reconstruction are presented and discussed.

II. Mathematical Background and Numerical Solution

In this section, we briefly present the mathematical background of the deformation modeling of the soft tissues. First, the physical formulation of the problem is sketched and then, the numerical solution is described.

A. Physical Representation of Deformations

Since the realistic behavior of the model is one of the key requirements in surgical simulation, the relation between nodal displacements of deformable body and applied surface forces is derived based on the theory of elasticity. Denoting an arbitrary particle in a deformable body as $\mathbf{x}$, the deformation is defined as a mapping $\varphi : \Omega \rightarrow \Omega'$ from the undeformed to the deformed domain given as $\varphi(x_i) = x'_i = x_i + u_i$ where u_i are components of displacement each defined as a function $u_i(x_1, x_2, x_3)$.

The important entity which can be regarded as internal measure of the strain is represented by the Green-StVenant strain tensor γ_{ij} which is defined as

$$\gamma_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i} + u_{m,i}u_{m,j}) \tag{1}$$

where the notation convention $u_{i,j} = \partial_j u_i = \frac{\partial u_i}{\partial x_j}$ for derivatives together with the summation convention are applied.

Throughout this paper, the static equilibrium between the external and internal forces in the deformed configuration Ω' is considered. The former are usually represented by the surface

traction forces $\mathbf{g}'$ and applied body forces $\mathbf{f}'$, whereas the latter are measured by *Cauchy stress tensor* τ' which represent the forces per unit area. Based on the conservation laws it can be shown that the static equilibrium of the deformed domain Ω' in the Eulerian system is represented by

$$\mathbf{0} = \mathbf{f}'(\mathbf{x}') + \nabla \cdot \tau'(\mathbf{x}') \qquad \mathbf{x}' \in \Omega' \tag{2}$$

$$\mathbf{g}'(\mathbf{x}') = \mathbf{T}'(\mathbf{x}', \mathbf{n}') \qquad \mathbf{x}' \in \partial\Omega' \tag{3}$$

where $\partial\Omega'$ is boundary of the domain Ω', $\mathbf{n}'$ is surface normal in $\mathbf{x}'$ and $\mathbf{T}'$ is Cauchy stress vector related to the Cauchy stress tensor via $\mathbf{T}' = \boldsymbol{\tau}'\mathbf{n}'$. The above equations can be transformed from the deformed coordinate system which is not a priori known to the undeformed coordinate system by Piola transformations. Then, the equilibrium equation can be formulated using the symmetric *second Piola stress tensor* σ_{ij} as

$$-f_i = \partial_j \left\{ (\delta_{im} + u_{i,m})\sigma_{mj} \right\} \tag{4}$$

where the summation convention is applied. The connection between the strain tensor γ_{ij} and the stress tensor σ_{ij} is provided by the constitutive law. In this paper, we focus on *hyperelastic* materials which are usually used for the soft-tissue modelling. The hyperelastic materials are characterized by the concept of *stored energy function* W, a scalar function which fully determines the stress/strain relationship. The function W is coupled with the second Piola stress tensor as

$$\sigma_{ij} = \frac{\partial W}{\partial \gamma_{ij}}. \tag{5}$$

There are multiple definitions of W suitable for various types of materials. In our study, we implement and compare the *St.Venant-Kirchhoff material* with Lamé coefficients λ and μ defined in terms of the strain tensor γ as

$$W = \frac{\lambda}{2}\gamma_{ii}\gamma_{jj} + \mu\gamma_{ij}\gamma_{ji} \tag{6}$$

and the *Mooney-Rivlin* material with two material constants C_{10} and C_{01}

$$W = 2\left[\gamma_{ii}(C_{10} + 2C_{01}) + 2C_{01}[\gamma_{ii}\gamma_{jj} - \gamma_{ij}\gamma_{ji}]\right] \tag{7}$$

where the non-linear incompressibility condition $det(\delta_{ij} + \partial_j u_i) = 1$ must be applied.

In the case of StVenant material combined with the large-strain tensor γ_{ij}, the resulting law is referred as being non-linear in geometry and linear in material, whereas if the Mooney-Rivlin stored energy with the incompressibility conditions is used, the resulting law is regarded as non-linear in geometry as well as the material.

B. Finite Element Formulation and Numerical Solution

The governing system introduced in the previous section consists of the partial differential equation given by Eq. 5 which is to be solved over some domain Ω which can be irregular and complex. Such a problem is usually solved by *finite element method* which consists of several steps. First, a weak formulation of the problem is derived based on multiplication of both sides by weight functions ψ_i and integration over the domain Ω, resulting in

$$\int_{\Omega} \left\{(\delta_{im} - u_{i,m})\sigma_{mj}\right\} \psi_{k,j} dV - \int_{\Omega} f_i \psi_k \; dV - \int_{\partial\Omega} g_i \psi_k \; dS = 0. \tag{8}$$

where also the boundary conditions from Eq. 2 are included.

Then, the discretization of the domain Ω is constructed using some simple-shaped elements such as tetrahedra. Each element is equipped with a complete set of shape functions $\{\phi_i\}$ which are used for two purposes. First, the unknown function u is interpolated as $u \cong \sum_i \phi_i \bar{u}_i$ where $\bar{u}_i$ are scalar coefficients representing the unknown variables. Further, the shape functions are used also as the weight functions ψ_i in the weak given by Eq. 8. Putting this together, the weak form can be reformulated over a typical element Ω^e resulting in a local system $\mathbf{K}^e(\bar{\mathbf{u}}^e) = \mathbf{f}^e$ where $\mathbf{K}^e$ is non-linear local stiffness matrix which depends on the actual value of $\bar{\mathbf{u}}^e$. The global system $\mathbf{K}(\bar{\mathbf{u}}) = \mathbf{f}$ is then assembled from the contributions of the elements.

The system is non-linear due to the non-linearity of the underlying PDE and cannot be solved directly. In this paper, iterative solution method combining the incremental loading and Newton-Raphson methods is considered. First, let's assume, that the deformation for some given load $\mathbf{F}$ is to be computed. Then, an *incremental loading path* $\{(\mathbf{0}) = \mathbf{F}_{(0)}, \mathbf{F}_{(1)}, \ldots \mathbf{F}_{(M)} = \mathbf{F}\}$ is constructed and sequence of non-linear systems $\mathbf{K}(\bar{\mathbf{u}}_{(i)}) = \mathbf{F}_{(i)}$ with initial estimation $\bar{\mathbf{u}}_{(i-1)}$ is solved for $i = 1, \ldots, M$. The solution in each step is obtained by *Newton-Raphson method*: having some estimation vector $\bar{\mathbf{u}}^{(j)}$ of the solution in the j-th iteration of the method, correction $\boldsymbol{\Delta}^{(j+1)}$ is computed solving linear system and applied to obtain a new estimation of $\bar{\mathbf{u}}^{(j+1)}$

$$\mathbf{K}'(\bar{\mathbf{u}}^{(j)})\boldsymbol{\Delta}^{(j+1)} = \mathbf{f} - \mathbf{K}(\bar{\mathbf{u}}^{(j)}) \tag{9}$$

$$\bar{\mathbf{u}}^{(j+1)} = \bar{\mathbf{u}}^{(j)} + \omega\boldsymbol{\Delta}^{(j+1)} \tag{10}$$

where $\mathbf{K}'$ is global tangent stiffness matrix which can be again assembled from the local tangent stiffness matrices $\mathbf{K'^e}$ computed for each element Ω^e. The scalar factor ω is obtained by line-search procedure in order to improve the convergence of the method by restricting the step size

in each iteration. There are two criteria for the Newton-Raphson method to stop: first, if the condition $\|\mathbf{f}-\mathbf{K}(\bar{\mathbf{u}}^{(j)})\| < \varepsilon$ holds in $j-$th iteration for some chosen ε, the calculation converged and the resulting solution is considered to be a valid data. Second, the iteration process is stopped after N iterations also if the condition above does not hold. In this case, the calculation is non-convergent and the data are considered to be invalid.

It is important to emphasize that before the solution of the Eq. 9 takes place, the Dirichlet boundary conditions (BC) must be imposed. Basically, there are two main types of the conditions: homogeneous and non-homogeneous boundary conditions. In the first case, the zero displacements are prescribed for some components of the unknown vector $\bar{\mathbf{u}}$, i. e. $\bar{u}_i = 0$ for some i. These conditions fixing the system in the space are essential for the uniqueness of the solution. Besides, non-homogeneous conditions $\bar{u}_j = b_j$ can be prescribed for some components of the vector $\bar{\mathbf{u}}$. The non-homogeneous conditions can be regarded as another "input" of the method, since although no external forces are applied, the body gets deformed if non-zero deformation is prescribed for some part of the body (e. g. one node).

There are several techniques for imposing the Dirichlet boundary conditions such as elimination, penalization or Lagrange multipliers. Among the three, the Lagrange multipliers can be used straightforwardly for obtaining the response of a node with prescribed displacement. Since the force is of a great importance for our application, the boundary conditions are imposed via Lagrange multipliers. The method based on the Lagrange multipliers is also utilized for the implementation of the incompressibility conditions introduced in the previous section. In this case, the system is also augmented and the resulting multipliers can be understood as pressure in the nodes of the mesh.

III. Integration of Soft Tissue Material Properties into Model

One of the main obstacles in developing realistic soft tissue models is the lack of data on the material properties of live organ tissues. Measuring and characterizing in vivo organ properties is a highly challenging task, but is a requirement for development of realistic surgical simulators. Soft tissue models with incorrect material properties will adversely affect training in VR-based surgical simulator systems. The research on tissue mechanics is extensive, but most of the earlier experiments took place in a laboratory environment (in vitro studies) under well-defined boundary and loading conditions. Typically, tissue samples taken from an organ of interest are transferred

to a laboratory in a chemical solution for measurement. Because researchers carefully decide on the sample geometry and experimental conditions in advance, they can easily obtain stress and strain values from the measurement data. However, mechanical properties of soft tissues change with time and the results obtained through in vitro measurements do not represent actual tissue properties.

In the past, we developed a robotic indenter for minimally invasive measurement of live tissue properties in a living body [17]. This system includes a robotic arm (Phantom haptic device from Sensable Technologies, model 1.0), a force sensor (Nano 17 from ATI Industrial Automation), and a long probe that has a round tip with a 2 mm radius. Using the robotic indenter, we performed a) static indentation and b) stress relaxation experiments on three pig livers and successfully measured the nonlinear and viscoelastic material properties of pig liver. An effective elastic modulus of pig liver was estimated from the static indentation data using the linear elastic contact theory and the small deformation assumption. In addition, an inverse FE solution was developed using ANSYS FE package to estimate the optimum values of nonlinear hyper-elastic material properties of pig liver via iterations. Hyper-elastic behavior of liver was modeled using 2-term Mooney-Rivlin strain-energy function defined by Eq. 7 in section II-A and the material coefficients C_{01} and C_{10} are estimated from the inverse solution.

In this study, the linear elastic modulus of pig liver that is estimated from the static indentation data (E = 15.48 kPa) is utilized in the linear St.Venant material model to calculate the Lamé coefficients and the non-linear material coefficients estimated via the inverse FE solution (C_{01} = 1.28 kPa, C_{10} = 1.3 kPa) are utilized in Mooney-Rivlin material model.

IV. Haptic Simulation Based on Precomputations

A. Simulation Parameters

In this section, the parameters affecting the interaction, i.e. the overall behaviour of the simulation model are presented and briefly described. In the following, the displacement-driven interaction is considered when the actual position of the haptic device (Haptic Interaction Point, HIP) is acquired via the haptic device driver and after the reaction force is computed, it is delivered back to the device to convey the sense of touch to a user of the system. This implies that from the computational point of view, the position is taken as the input and force is calculated and returned as the output of the computations.

Let's assume a deformable body with a domain Ω that has been discretized resulting in the mesh $\mathcal{M}$ given by the set $\mathcal{V}$ of the vertices (nodes) and set $\mathcal{E}$ of the elements. First, the behaviour observed during the interaction is determined by physical parameters. These are mainly the material coefficients defined for the selected model. Besides, there can be other parameters representing some external conditions; the amplitude of an external force (e.g. gravitation) can be considered as physical parameter of the simulation. Second, there are geometric parameters which are related to the mesh of the deformable body. The first geometric parameter is represented by a set $\mathcal{F} \subset \mathcal{V}$ of surface nodes which are fixed in space during the interaction. From the numerical point of view, the set $\mathcal{F}$ defines the *homogeneous Dirichlet conditions* setting the components of the nodes from $\mathcal{F}$ to zero.

As the displacement driven interaction is being considered, there is at least one surface node, for which a non-zero displacement is prescribed. In this paper, single-point interaction is studied: when the HIP penetrates into the object, a single *active node* $\mathcal{A}$ on the surface of the body is selected. During the simulation, the active node is associated with the components of its actual displacement. These values represent the last parameter which is, unlike the parameters introduced before, updated in each iteration of the haptic loop depending on the actual position of HIP. The prescribed displacement of $\mathcal{A}$ is therefore the *control parameter* of the simulation and from the numerical point of view, it can be regarded as non-homogeneous Dirichlet condition.

Ideally, the computational model of the single-point static-equilibrium interaction works as follows: given the mesh of the deformable body, all the physical parameters of the model are specified together with the set $\mathcal{F}$ of the fixed nodes. Then, the interaction starts and in each iteration of the haptic loop, the instant position of HIP is acquired and the collision detection between HIP and the geometric surface model is performed. If the collision occurs, the surface node which is closest to the HIP position is selected as active and it is coupled with the HIP. From now on, in each iteration of the haptic loop, the control parameter (i.e. the actual position of the active node) is updated and the corresponding static equilibrium is calculated resulting in the overall deformation of the body and reacting force.

It is important to recall that the calculation of the static equilibrium of a complex body using FEM and prescribed displacement as the control parameter is a computationally expensive iterative process as shown in the mathematical background section. These iterations cannot be performed inside the haptic loop running at high update rate high enough for stable haptic

interaction. Therefore, an approach based on precomputation of configuration spaces is presented for real-time interaction in the following section.

B. Configuration spaces: Discretization and Approximation

After specifying the parameters of the interaction, notion of configurations and configuration spaces are now introduced.

For a given set $\mathcal{F}$ of the fixed nodes and active node A, the actual state of the simulation can be completely described by configuration C consisting of the current position $\mathbf{p}$ of HIP (the control parameter), vector $\mathbf{u}$ of displacements and force $\mathbf{h}$ acting on the active node.

As the control parameter changes from $\mathbf{p}$ to $\mathbf{p}'$, the new reaction force $\mathbf{h}'$ and deformation $\mathbf{u}'$ of the body are computed so that the static equilibrium is restored. In other words, a transition from the previous configuration C to the new configuration C' occurs: $C \rightarrow C'$. The set of all possible configurations, computed while having node $\mathcal{A}$ as active, is denoted as $\mathcal{S}_{\mathcal{A}}$. Now, the haptic interaction can be regarded as travelling through configuration space $\mathcal{S}_{\mathcal{A}}$ where each step is defined by a transition from one configuration to another.

Obviously, the configuration space $\mathcal{S}_{\mathcal{A}}$ is continuous and infinite. The main idea of the method is to discretize $\mathcal{S}_{\mathcal{A}}$ by constructing a finite subspace $\mathcal{D}_{\mathcal{A}} \subset \mathcal{S}_{\mathcal{A}}$ that can be efficiently precomputed and moreover, each configuration $C \in \mathcal{S}_{\mathcal{A}}$ can be approximated by some fast procedure using only the data from $\mathcal{D}_{\mathcal{A}}$. Thus, the approach proposed in this paper can be summarized as follows:
Off-line precomputation phase. For a given FE mesh, all the physical parameters are specified and both the set $\mathcal{F}$ and the active node $\mathcal{A}$ are selected. Then, a set of points $\mathcal{G}_{\mathcal{A}}$ is chosen somehow in the space surrounding the rest position of the active node $\mathcal{A}$ and for each such a point $\mathbf{g} \in \mathcal{G}_{\mathcal{A}}$, the active node is displaced to the position $\mathbf{g}$ and the corresponding configuration is computed and stored. Details on this process are presented in section IV-C.

On-line interaction phase The precomputed configurations from the discrete set $\mathcal{D}_{\mathcal{A}}$ are used to approximate an arbitrary configuration $\tilde{C}$ associated with the current position $\mathbf{p}$ of HIP. The approximation is calculated by interpolation of the precomputed data which is fast enough to be performed inside the haptic loop. The details of the interpolation phase are presented in section IV-D.

C. Off-line Precomputations

In this section, the selection of the points $\mathcal{G}_\mathcal{A}$ from the previous section is revealed and the algorithm how to construct the corresponding set $\mathcal{D}_\mathcal{A}$ is presented.

The algorithm presented in this section is based on the assumption that the solution of the boundary problem solved in each step of the simulation is unique. This assumption is based on the ellipticity of the boundary problem being solved in each step of the simulation. This assumption combined with the single-point interaction implies that there is at most one configuration C corresponding to the particular position $\mathbf{p}$ of the HIP.

By virtue of the assumption about the uniqueness of the solution, it is sufficient to define the discretization structure $\mathcal{G}_\mathcal{A}$ as a *uniform grid* of points surrounding the rest position $\mathbf{x}_0^\mathcal{A}$ of the active node $\mathcal{A}$ which coincides with one of the grid points. So for $i, j, k \in \mathbb{Z}$, the grid is defined as

$$\mathcal{G}_\mathcal{A} = \{\mathbf{g}_{ijk} \in \mathbb{R}^3 | \mathbf{g}_{ijk} \leq r_\mathcal{G} \wedge \mathbf{g}_{000} = \mathbf{x}_0^\mathcal{A}\}. \tag{11}$$

There are two parameters of the grid. The first one is represented by the radius $r_\mathcal{G}$ determining how far the active node is going to be displaced from the rest position. The second parameter sets the density of the grid, i. e. the distance between two adjacent points of the grid.

The construction of the corresponding set $\mathcal{D}_\mathcal{A}$ can be defined recursively as follows.

1) For the point $\mathbf{g}_{000}$ corresponding to the rest position of the active node, store the zero configuration C_{000} where all the displacements of the nodes as well as the reacting forces are set to zero.
2) If a configuration C_{ijk} has been already computed and stored in the point $\mathbf{g}_{ijk}$, compute the transitions $C_{ijk} \rightarrow C_{i'j'k'}$ to all adjacent positions $\mathbf{g}_{i'j'k'}$, such that $\|i - i', j - j', k - k'\| \in \{1, \sqrt{2}, \sqrt{3}\}$ provided $C_{i'j'k'}$ has not been already computed. This step can be regarded as a *configuration expansion*.

   The particular transition $C_{ijk} \rightarrow C_{i'j'k'}$ is constructed by the incremental loading technique: the path between $\mathbf{g}_{ijk}$ and $\mathbf{g}_{i'j'k'}$ is split into M steps and C_{ijk}^0 is set to C_{ijk}. Then, the path is traversed by executing Newton-Raphson method to compute the configuration C_{ijk}^m from the initial state C_{ijk}^{m-1} for $m = \{1, \ldots M\}$.

   The intermediate configurations can be stored as well since they can be utilized by the interpolation methods used in this study to interpolate forces.

3) Repeat the step 2 until for each point $\mathbf{g}_{ijk} \in \mathcal{G}_{\mathcal{A}}$ there is the corresponding configuration C_{ijk} computed and stored.

It is to be emphasized that the configuration space will not cover all the points from the grid $\mathcal{G}_{\mathcal{A}}$ (i. e. for some of the points in the grid, the corresponding configurations will not be computed). There are two main reasons for this:

1) Haptic devices have a limitation concerning the maximal force $\mathbf{f}_{max}$ it can display. This implies that if the reaction force $\mathbf{f}$ in some configuration C_{ijk} exceeds $\mathbf{f}_{max}$, the configuration is not reachable since the limitation of the device does not allow the user to reach this position by HIP. Therefore, such a configuration is not expanded any more and the configurations corresponding to the points beyond $\mathbf{g}_{ijk}$ are not computed.
2) The construction of each step within a transition $C_{ijk} \rightarrow C_{i'j'k'}$ is an iterative process which can fail numerically. This can happen for several reasons; either the deformation is too large to be modelled physically, or some elements of the mesh get degenerated causing the distortion of the mesh or the chosen material is not suitable for that type of deformation. If such a *failure point* occurs on the path from $\mathbf{g}_{ijk} \rightarrow \mathbf{g}_{i'j'k'}$, the configuration $C_{i'j'k'}$ is not inserted to the state space (i. e. there is no configuration for the point $\mathbf{g}_{i'j'k'}$).

D. On-line Interaction

In this section, the interpolation of the reaction force $\mathbf{h}$ which is computed during the real-time haptic interaction phase is explained. The nodal displacements $\mathbf{u}$ is computed analogically for the visualization purposes. Each component of $\mathbf{h}$ is interpolated separately using the precomputed force data. In the following, two different methods are considered: polynomial and radial-based interpolation (see survey [18]). For each of them, linear and cubic versions are employed.

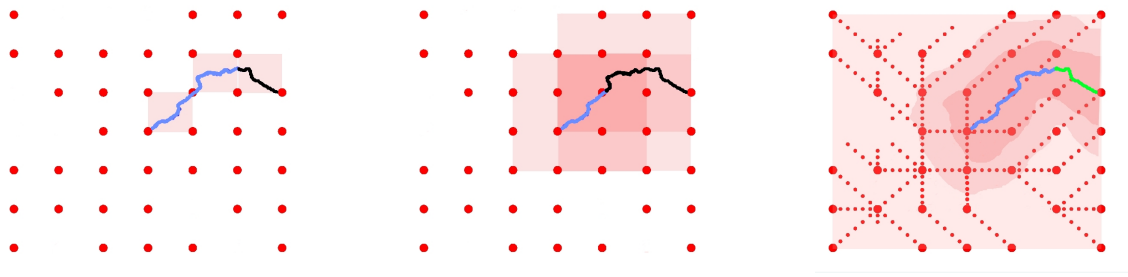
The polynomial interpolation works with uniformly distributed grid data, i. e. only the precomputed configurations C_{ijk} stored in the points of the grid $\mathcal{G}_{\mathcal{A}}$ can be utilized. Let's assume that in actual iteration of the haptic loop, the actual position $\mathbf{p}_{\mathcal{A}}$ of the currently active node $\mathcal{A}$ is acquired from the haptic device. For the simplest case of the linear interpolation, the cell of the grid $\mathcal{G}_{\mathcal{A}}$ where the position $\mathbf{p}_{\mathcal{A}}$ is located is identified by computing the eight neighboring points which are the corners of the cell. Afterwards, the components of the force vector within the configurations stored in the selected grid points are used for the tri-linear interpolation. The technique is illustrated by the Fig. 1(a). Clearly, the tri-linear interpolation is fast enough,

however, it becomes unreliable as the nonlinearities in the model dominate the force response. The solution can be provided by interpolating from finer grid, that can, however, significantly increase the number of precomputations. It is also important to emphasize that the interpolation within a cell can be computed provided all eight nodes of the cell are available. Therefore, as soon as the current position $\mathbf{p}_{\mathcal{A}}$ gets out of the grid, the configuration cannot be approximated well.

The next alternative within the frame of the polynomial method is *tri-cubic interpolation*. In this case, besides the cell containing the actual position $\mathbf{p}_{\mathcal{A}}$, eight additional neighbouring cells are selected resulting in a *super-cell* with 64 grid points providing the configurations for the interpolation (see Fig. 1(b). In this case, computations are more demanding, but still can be done within the haptic loop. A more serious drawback is caused by the fact that the interpolation can be computed only in the case when all the configurations for the 64 points in the actual supercell are available. For a given grid of precomputed configurations, the tri-cubic interpolation may not work close to the border of the grid or in the case where the points are missing due to the non-convergent calculations. Therefore, it can be applied in smaller region close to the rest position of the active node.

Regarding the complexity of the polynomial interpolations, the number of floating-point operations for both tri-linear and tri-cubic polynomial interpolations are independent of the size of the grid: in the former case, about 250 floating-point operations are needed to compute one component of the force, whereas 2,000 operations are necessary for the tri-cubic method. When comparing to the performance of today's processors, the interpolation of the force vector can be easily computed within the haptic loop.

For further improvement in the accuracy of the approximation, a *radial-based function* (RBF) depicted by Fig. 1(c) is employed for the interpolation. It is suitable for irregularly scattered data, so beside the configurations C_{ijk} stored in the points of the grid $\mathcal{G}_{\mathcal{A}}$, the intermediate states C^m_{ijk} can be utilized as well. In this case, it is not necessary to localize the actual position $\mathbf{p}_{\mathcal{A}}$ within the grid $\mathcal{G}_{\mathcal{A}}$ because in each step, all the precomputed data will be used with different weights depending on the distance from the actual position $\mathbf{p}_{\mathcal{A}}$. Briefly, having a set of positions $\mathbf{p}_i$ in space together with the corresponding values $\mathbf{h}_i$ stored in those positions, the interpolated



(a) (b) (c)

Fig. 1. A 2D illustration of the three interpolation methods discussed in this paper: (a) bi-linear interpolation (b) bi-cubic interpolation and (c) radial-base function interpolation.

value $\mathbf{h}$ at some position $\mathbf{p}$ is computed as

$$\mathbf{h} = \sum_{i=1}^{N-1} w_i \phi(\|\mathbf{p} - \mathbf{p}_i\|) \tag{12}$$

where ϕ is a user-defined function referenced as the *kernel*. In our study, $\phi(r) = |r|$ and $\phi(r) = r^3$ are used for linear and cubic interpolation, respectively. The weights w_i can be calculated during the off-line phase of the computation by solving the linear system which is assembled after substituting the known values of $\mathbf{h}_i$ into the Eq. 12.

Therefore, the technique consists of two phases — computation of the weights and interpolation of the actual value. Each phase of the computation has different complexity. Let's first focus on the computation of the interpolated value in the second phase. Comparing to the tri-linear and tri-cubic interpolation, the time complexity of this step linearly depends on the number of nodes in the grid. In other words, it is $\mathcal{O}(\sqrt[3]{N})$ for the three-dimensional grid $r_\mathcal{G}$. This can be computed in real time for grids having from hundreds to thousands of grid points, depending on the performance of the computer running the haptic loop. The first step is computationally much more expensive, this time the complexity is $\mathcal{O}(N)$. Nevertheless, this computation is not performed in the haptic loop, but the calculation of the weights can be performed as part of the precomputation process together with the construction of the discretization space. In this case, also the interpolation is smooth for any motion of the HIP, since the solution depends on all the nodes of the grid always. The method can be used for the interpolation also close to the border of the uniform grid. Moreover, it can be still applied successfully even when there are some configurations missing. However, the accuracy of the solution is not guaranteed anymore

and the results of the approximation may become physically invalid.

V. Implementation and Computations

A. Distributed Implementation of the Space Precomputation

First, we briefly focus on the implementation of the precomputation phase. As stated before, all the deformations considered in this paper have been computed by physically-based finite element method. For this purpose, freely available library GetFEM [19] has been employed for assembling the stiffness matrices. The solution is obtained by incremental staging when the control parameter (i. e. the prescribed displacement of one node of the FE mesh surface), is changed in small steps. In each step, the Newton-Raphson method with line search is applied to minimize the residual. In each iteration of the Newton method, the linearized system is solved using the MUMPS linear solver [20].

The construction of the configuration space has been implemented recursively as described in the section IV-C. In order to speed up the precomputations, the independent calculations are done in a distributed manner on a cluster of computers. A centralized approach is used for the implementation, when there is a central client process working as a very simple *scheduler* distributing the work among working processes, *servers*. There are two tables maintained by the scheduler. First, it is the table of idle servers which are available. Initially, all the servers are inserted to the table. Second, it is the table of all the configurations which are to be computed. Each is marked by exactly one of the following tags

- **unknown:** the configuration has not been computed so far, only its location within the grid is known;
- **expandable:** the configuration has been successfully computed and can be used as initial estimation for further computations;
- **terminating:** the configuration has been successfully computed, but it is not reachable by the haptic device, so it is not necessary to perform further computations;
- **failed:** the computation of the configuration has failed due to the convergence issues.

Initially, the zero configuration corresponding to the rest position of the active node is marked as expandable and all other configurations are marked as unknown. The configuration space is then constructed as follows: if there is at least one idle server, the scheduler chooses some unknown configuration C' for which there is an expandable configuration C such that C and C'

are adjacent in the grid. Afterwards, the computation of the transition $C \rightarrow C'$ is assigned to the idle server which is removed from the table of idle servers.

The computation of transition can lead to three different results regarding the tag of the resulting configuration. First, if the iterative process does not fail, then the resulting configuration is valid and the magnitude of reaction force is compared to the force limit $\mathbf{f}_{max}$. In the case when the force is lower than this limit, the configuration is marked as expandable, since it is reachable during the interaction and further configurations can be computed using this one as initial estimation. In the opposite case, the configuration is marked as terminating, because it is not necessary to expand it further since it is not reachable by the haptic device. In both cases, the configuration is stored to be used for the interpolation process. Second, if the computation fails, the resulting configuration is marked as failed. In this case it is not stored, since it cannot be used by the interpolation process. For all the cases enumerated above, the scheduler is notified by the server about the result of the computations, so that the server can update the tables: the tag of the new configuration is changed in the table of configurations according to the results of computations and the server is now idle. The computations are stopped, if there is no unknown configuration (i. e. all the configurations are marked as either expandable, terminating or failed).

B. Implementation of the Space Approximation

After giving the details of the precomputation phase, we now focus on the implementation of the interaction phase. We have simulated the real-time visio-haptic interactions between a point (i. e. HIP) and the 3D surface model of human liver (made of triangular patches). The hardware components of our simulation system include a computer monitor for visual display of deformations and a haptic device for simulating the force interactions between the HIP and the liver (Fig. 2). The underlying code is written in MS Visual C++ environment, the graphical rendering of the object and the visual deformations are displayed using Open Inventor (a scene graph API), and the haptic feedback to the user is provided via PHANToM haptic device (SensAble Technologies) using GHOST v.4.0 driver.

Our real-time computational architecture for simulating nonlinear tissue response consists of three threads running asynchronously: haptic, modeling, and visual thread. The haptic thread, updated at 1 kHz, acquires the new position of the haptic probe as the user manipulates the probe. The modeling thread, updated at 100 Hz, performs collision detection, determines the

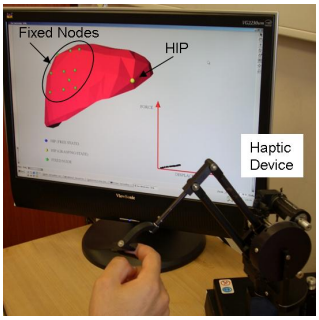


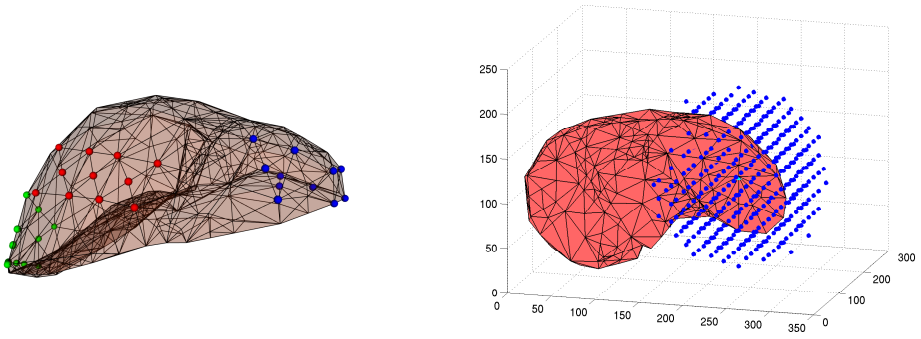
Fig. 2. The hardware components of our simulation system.

active node, and calculate the collision response based on the interpolation of the precomputed data stored at the grid points surrounding the active node. Finally, the visual thread, updated at 30 Hz, graphically renders the haptic interface point (HIP) and the deformations of the 3D model.

C. Model and Computations Overview

All the experiments discussed in the next paragraph of this section have been performed on the 3D model of human liver obtained from the INRIA repositories. The model has been meshed by TetGEN mesh generation tool [21], resulting in two meshes with 1777 elements (501 nodes) and 10280 elements (2011 nodes), respectively. The length of the model is about 22 cm. For the experiments, two different sets of fixed points $\mathcal{F}_1$ and $\mathcal{F}_2$ are tested, fixing the area on the left and top part of the organ, respectively. Further, 10 different active nodes are selected for the testing purposes. Both parameters are depicted in Fig. 3(a).

Regular and random loading paths are generated and corresponding configuration spaces are computed. Regarding the regular paths, 26 paths are used for ten active nodes and two sets $\mathcal{F}_1$ and $\mathcal{F}_2$ of fixed nodes so in total, 520 ($26 \times 10 \times 2$) of them are tested for each material type. Each path starting in some active node $\mathcal{A}_i$ is 10 cm long and the length of each step in the incremental loading is 1 mm to improve the convergence of the Newton-Raphson method. The construction of a single path takes from 15 to 30 minutes depending on the type of material and convergence rate. Since more than 150,000 configurations has been constructed in total, the regular path generation has been performed on a cluster having 32 CPU cores (Intel Xeon at 3.0 GHz and AMD Opteron at 2.8 GHz) and it takes about 60 hours to compute all configurations.



(a) (b)

Fig. 3. (a) The liver model used for the experiments. The nodes are coloured according to the functionality: the green and red nodes belong to the fixed-node sets $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively. The blue nodes belong to the set of the active nodes. (b) Illustration of a grid used for the precomputation of configuration spaces.

The results have been stored on shared disk arrays for the inspection and interpolation. Regarding the random paths, for each material about 5000 paths are tested. Each path consists of 200 steps and incremental step size is 1 mm as well. In total, about 3 million configurations are constructed. The computations of random paths takes about five days on cluster of 48 CPUs.

Four uniform grids with different densities has been generated and the corresponding configurations has been computed. Each grid is enclosed in a sphere with radius 10 cm for each active node being the center of the sphere. The points inside the grid are distributed regularly dividing the volume of the sphere into cubic cells. The cell sizes are 20 mm, 14.3 mm, 10 mm and 6.667 mm, the corresponding grids had 514, 1418, 4168 and 14146 points respectively. The grid with 514 points is illustrated in Fig. 3(b).

VI. Evaluation and Discussion

A. Behaviour of the Model

In the following, the behaviour of the liver model is briefly studied using force-response plots obtained from the application of the regular loading paths introduced in the section V-C. First, the stability of the computations is analyzed, as the non-linear systems are considered. Second, the force-displacement curves are briefly studied to show the importance of the parameters introduced in the section IV-A.

The detection of the non-convergent computations during the construction of the configurations is crucial for the evaluation of the method presented in this paper, since the interpolation cannot

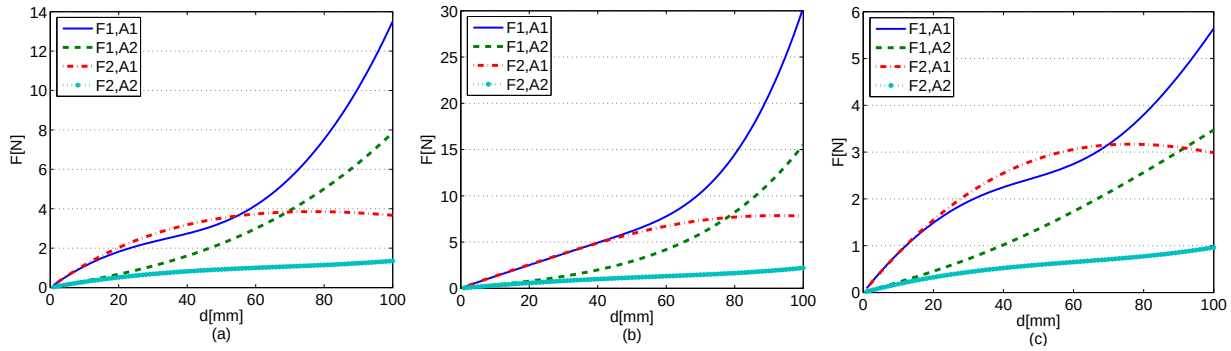


Fig. 4. Force displacement curves for the combinations of the fixed nodes sets ($\mathcal{F}_1$, $\mathcal{F}_2$) and active nodes (A_1,A_2). The results are depicted for three materials: (a) StVenant-Kirchhoff, (b) modified StVenant-Kirchhoff and (c) Mooney-Rivlin.

be applied successfully to invalid data produced by non-convergent computations (see section II-B). The tests using the regular paths shows that the convergence problems are mostly encountered for the StVenant-Kirchhoff material. In this case, 53% of the paths are non-convergent comparing to less than 5% in Mooney-Rivlin material.

Further investigation reveals that mainly the paths when compression-like deformation takes place are affected. The history of the mesh deformation is then studied more precisely for the non-convergent paths. It turns out that despite the fact that incompressibility is imposed implicitly for the StVenant material by setting Poisson ratio to 0.49, some elements become degenerated during the loading process. In order to avoid this behaviour, the explicit incompressibility conditions (see section II-A) are imposed also for the StVenant material resulting in much more stable behavior with 95% of convergent paths. Because of the good numerical properties of the StVenant material equipped with the incompressibility conditions, it is included in all the experiments presented in this paper being called as *modified StVenant material*.

Location of the fixed nodes and the contact node used for the manipulations (i. e. active node) are used as the control parameters to simulate the behavior of liver model for different material types. For the evaluation, two sets of fixed nodes $\mathcal{F}_1$ and $\mathcal{F}_2$ and two active nodes $\mathcal{A}_1$ and $\mathcal{A}_2$ are used (the forcer located closer to the gravity center of the body, the latter closer to the right "tip" of the liver. The force-displacement responses of the three materials for different combinations of the fixed nodes and the active nodes are given in Fig. 4. When comparing the shape of the curves within each graph, it is clear that selection of the boundary conditions (i. e. fixed and

active nodes) plays important role in force response. First, the concave curves has been recorded for the case when the set of the nodes $\mathcal{F}_2$ on the top of the liver was fixed. This selection of the fixed nodes is quite restrictive resulting in local deformations. Hence only the elements around the active node are deformed to follow the prescribed displacement. On the other hand, for the set $\mathcal{F}_1$ the liver can be deformed globally, since both active nodes are located on the opposite side w. r. t. the set $\mathcal{F}_1$. This results in convex force-response. It is observed that the magnitude of the response force seems to be determined by the location of the active node w. r. t. the body. For the of $\mathcal{F}_1$, the displacement of the node $\mathcal{A}_1$ results in larger response force, as it is closer to the gravity center. Putting it all together, the force-displacement curves recorded during the large number of the experiments with regular loading paths show the importance of the boundary conditions, i. e. location of the fixed and active nodes, on the force response of the soft body.

TABLE I

WALL TIME OF SINGLE NEWTON ITERATION.

No.	mat.	#nodes	#elems	order	#equations	assembly [s]	solve [s]	total [s]
1	SV	501	1777	lin.	1539	0.49	0.05	0.54
2	MR	501	1777	lin.	2040	1.05	0.23	1.28
3	SV	2011	10270	lin.	6069	3.22	0.71	3.93
4	MR	501	1777	quad.	12323	3.38	3.18	6.56
5	MR	2011	10270	lin.	8080	8.15	1.74	9.89

B. Construction of Configuration Spaces

Before discussing the computational cost of constructing configuration space, we first present the computational time required for one Newton-Raphson iteration employing two different materials, StVenant-Kirchhoff (SV) and Mooney-Rivlin (MR) and two meshes of different size: one with linear and another with both linear and quadratic elements (see Table I). The computational time for the modified StVenant material are close to those obtained for the Mooney-Rivlin material.

The computations have been executed sequentially on AMD Opteron Processor 250 (2 GHz) with 8 GB of physical memory and the execution times are calculated as the average of large number of iterations. The larger number of equations for the Mooney-Rivlin material shown in Table I is caused by the enforcement of the incompressibility conditions via the Lagrange multipliers. This also leads to longer time needed for the assembly process. In the case when

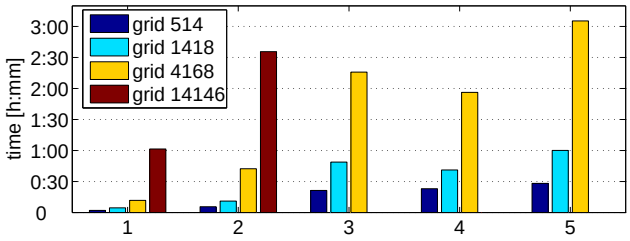


Fig. 5. The graph showing wall clock time needed to compute one state space on 64 CPU cores. Each group corresponds to one combination of parameters (indexed as rows in Tab. I) and the configuration spaces for the four grids are computed.

line search technique is applied, additional time is required in each iteration. Since the wall time spent by a single Newton iteration is shown, it must be multiplied by the number of iterations needed for the computation of one configuration. Therefore, the total time of the Newton method including the line search can result in more than one minute.

For this reason the configuration space is constructed using a distributed computing approach. All the experiments have been executed on 16 nodes of a cluster, each having $2\times$ Dual Core AMD Opteron Processor 270 with 8 GB of physical memory running at 2 GHz, so in total, 64 processor cores (16×4) have been used for the precomputation and the data is stored on local disks for further processing. The lengths of the computations are shown in Fig. 5. The configuration space can be constructed quickly for the grid with 514 points, ranging from 2 to 28 minutes depending on the complexity of the model. The computational cost of constructing the grid with 1418 points is still acceptable (about one hour for FE model having more than 10 thousand elements). The finer grid (4168 points) is affordable mainly for simpler models (a linear FE model having 501 nodes is computed in less than 45 minutes), but more complex models may take about 3 hours. Finally, the grid with 14146 points is tested only with two simple FE models, confirming the cubic increase of the computational time due to the increase in number of grid points.

C. Evaluation of the Space Reconstruction

In this section, the second part of the key results of this paper are presented to justify the validity of the proposed precomputation approach which is based on the reconstruction of the configuration spaces. This is performed by comparing the configurations computed precisely with

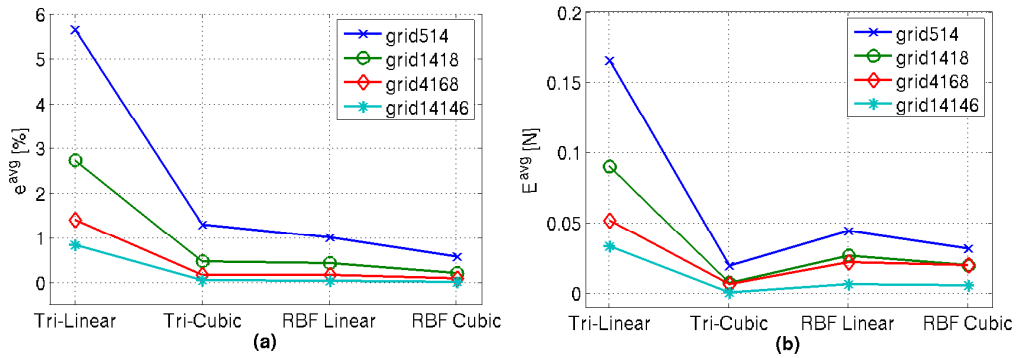


Fig. 6. Relative average error e^{avg} depicted in (a) and absolute average error E^{avg} depicted in (b) for the various grids and interpolation techniques.

their counterparts obtained by the interpolation of the precomputed data. For this purpose, both the regular and the random loading paths are utilized: for each configuration C computed during a particular loading, its counterpart $\tilde{C}$ is reconstructed by interpolation using the precomputed spaces. This reconstruction process is tested with four different interpolation methods: tri-linear and tri-cubic and radial-based function with linear and cubic kernels. Moreover, each interpolation method is tested with four precomputed uniform grids having different number of points (514, 1418, 4168 and 14146).

For the purpose of evaluating the interpolation accuracy, two error measures are defined and calculated for all random and regular loading paths for each interpolation method and the grid density. Both error measures are calculated using the components of the force (x, y and z in this case) instead of its magnitude in order to determine the accuracy more strictly. First, the absolute mean error showing the average difference between the interpolated force component $\tilde{f}^i_C$ and precisely computed force component $f^i_{\tilde{C}}$ of each configuration along a path $\mathcal{P}$ is calculated as:

$$E_{\mathcal{P}} = \frac{1}{3|\mathcal{P}|} \sum_{i \in \{x,y,z\}} \sum_{C \in \mathcal{P}} |f^i_C - \tilde{f}^i_{\tilde{C}}|. \tag{13}$$

Similarly, the relative mean error is computed as ratio of absolute mean error to the precisely computed force component $f^i_{\tilde{C}}$:

$$e_{\mathcal{P}} = \frac{1}{3|\mathcal{P}|} \sum_{i \in \{x,y,z\}} \sum_{C \in \mathcal{P}} \frac{|f^i_C - \tilde{f}^i_{\tilde{C}}|}{|f^i_C|}. \tag{14}$$

When small forces (e.g. bellow 0.5 N) are interpolated, the relative error can increase rapidly,

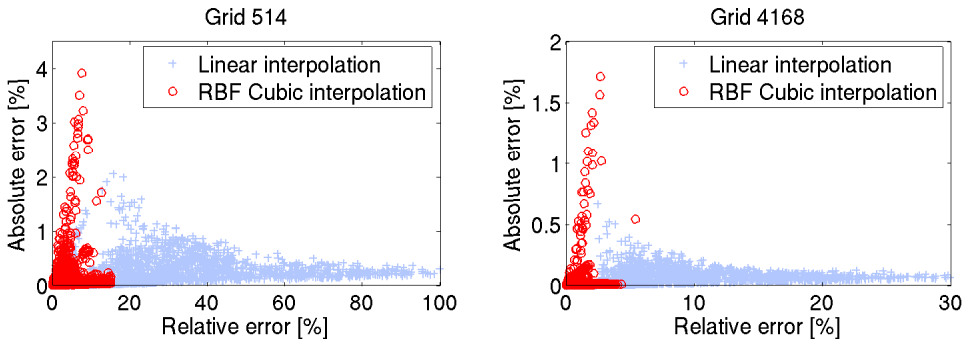


Fig. 7. Maximum errors of tri-linear and RBF-cubic interpolation: maximum error for each generated path is depicted by one circle and cross, respectively.

although the absolute difference is even bellow the resolution of the device or human haptic perception. For the case of larger forces (above 20 N), the absolute difference can become significant (over 1 N), but it is negligible w. r. t. the overall magnitude of the force. To be able to present the details in a compact way, the absolute and relative errors introduced above are further averaged over all the computed paths resulting in mean absolute error E^{avg} and mean relative error e^{avg} computed separately for each of the interpolation methods and the grid densities. The resulting values are presented in Fig. 6.

The graph shows that the influence of the interpolation on the accuracy is significant, since the accuracy obtained by the RBF interpolation with cubic kernel using the coarsest grid with 514 points is comparable to the accuracy of the tri-linear interpolation using the finest grid with 14146 points. The tri-cubic interpolation seems to show the best accuracy w. r. t. the absolute error. Nevertheless, it is important to recall that the tri-cubic interpolation can be used only in regions where complete supercell composed of 64 grid points is available (see section IV-D). E.g., for the grid having 514 points, only 31% of the configurations generated during the experiments can be interpolated using the tri-cubic method due to the missing data. Therefore, the utilization of tri-cubic interpolation is limited to smaller area closer to the rest position of the active node that leads to optimum absolute error, since larger forces corresponding to the boundary areas are not included in this type of interpolation.

Beside the average error of the interpolation methods, the worst case scenario is investigated using the maximum relative and absolute errors for each path denoted $e^{max}_{\mathcal{P}}$ and $E^{max}_{\mathcal{P}}$, respec-

tively. The definition of both quantities is similar to Eq. 13 and Eq. 14 except the summation signs are replaced by maximum signs. The results are shown in Fig. 7 where tri-linear and RBF cubic interpolations are presented for two grids having different densities. The graph shows that the maximum relative error (x axis) can be significantly reduced when using the RBF cubic interpolation instead of tri-linear. The figure shows that tri-linear interpolation on the dense grid (blue dots on the right-hand side) results in maximum relative error achieving 30%, whereas the RBF interpolation on the coarse grid (red dots on the left-hand side) results in maximum relative error under 20%. Moreover, the distribution of dots in both graphs show the relation between the absolute and relative errors. As the dots depicting the maximum errors are close to the axis, it can be concluded that although the relative error is large, the corresponding absolute difference is low or vice versa, the large absolute error occurs mainly when the absolute magnitude of the force is large and so the relative error is in fact low.

VII. Discussion and Future Work

Finally, a brief discussion of the results obtained during the evaluation is given bellow.

The main advantage of the precomputation approach proposed in this paper that any computationally expensive calculations are performed off-line, independently from the real-time haptic interaction. Therefore, complex models with non-linear material can be simulated with no impact on the stability of the haptic interaction. The main limitation of the proposed approach is represented by the computational time required for the precomputations. Also, fixing the parameters of the model in advance can be inconvenient, as for any modification on them causes the configuration space to be updated. Nevertheless, the length of the precomputation phase can be significantly shortened using a distributed computing approach suitable for the heterogeneous and geographically distributed environment which is the characteristic of computational grid architectures. Generally, the more computational resources, the more complex models can be studied.

Concerning the reconstruction phase, the simple tri-linear interpolation can be acceptable, if combined with denser grid. The tri-cubic interpolation displayed better properties w. r. t. the accuracy measures also on the sparser grids, but this method can be used only in limited region where all the 64 points required for the interpolation of a configuration. Nonetheless, both polynomial methods can be combined, so that the tri-cubic interpolation is used in the area

close to the rest position of the active node, where sufficient data are available, whereas the tri-linear method is applied in the regions close to the border of the grid. The RBF method showed to be the best choice when the accuracy is important, as it still can be computed within the haptic loop, provided that the weights are computed within the precomputation phase. It was shown that the cubic variant of RBF applied to the sparsest grid, can give better results in terms of the accuracy than the tri-linear method applied to the densest.

Concerning the future work, we are currently working on a solution which does not depend on the time consuming precomputation phase. The idea is based on the fact that the motion of HIP controlled by the user is relatively slow. Therefore, if the trajectory followed by HIP is roughly estimated based on the movements of HIP, the possible configurations surrounding the trajectory can be computed in advance and the reaction force corresponding to the actual position of HIP can be interpolated from the precomputed subspace. Hence, the whole configuration space is not constructed again, but only a part of it is updated. In order to generate the configurations in real time, GPU accelerators are used. We believe that this approach based on local configuration spaces can handle topological changes dynamical effects, which are both of great interest in the area of surgical simulations.

ACKNOWLEDGMENT

The first and last authors acknowledge the financial support provided by Ministry of Education, Youth and Sport of the Czech Republic under the research intent number 102/05/H050. The second and the third authors acknowledge the financial support provided by TUBITAK under contract number MAG-104M283 and the student fellowship program BIDEP-2210 The access to the METACentrum computing facilities provided under the research intent MSM6383917201 is acknowledged.

REFERENCES

[1] C.Basdogan, M.Sedef, M.Harders, and S.Wesarg, “Virtual reality supported simulators for training in minimally invasive surgery (invited paper),” *IEEE Computer Graphics and Applications*, vol. 27, no. 2, pp. 54–66, 2007.

[2] M. Bro-Nielsen and S. Cotin, “Real-time volumetric deformable models for surgery simulation using finite elements and condensation,” *Computer Graphics Forum*, vol. 15, no. 3, pp. 57–66, 1996.

[3] G. Székely, C. Brechbühler, R. Hutter, A. Rhomberg, and P. Schmid, “Modelling of soft tissue deformation for laparoscopic surgery simulation,” *Medical Image Analysis*, pp. 57–66, March 2000.

[4] C. Basdogan, C. Ho, and M.A.Srinivasan, "Virtual environments for medical training: Graphical and haptic simulation of common bile duct exploration," *IEEE/ASME Transactions on Mechatronics*, vol. 6, no. 3, pp. 267–285, 2001.

[5] D. C. Popescu and M. Compton, "A model for efficient and accurate interaction with elastic objects in haptic virtual environments," in *GRAPHITE '03: Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, (New York, NY, USA), pp. 245–250, ACM Press, 2003.

[6] S. Cotin, H. Delingette, and N. Ayache, "Real-time elastic deformations of soft tissues for surgery simulation," *IEEE Transactions On Visualization and Computer Graphics*, vol. 5, pp. 62–73, January-March 1999.

[7] G. Picinbono and J.-C. Lombardo, "Extrapolation: a solution for force feedback?," in *International Scientific Workshop on Virtual Reality and Prototyping*, (Laval France), pp. 117–125, June 3-4 1999.

[8] G. Picinbono, H. Delingette, and N. Ayache, "Non-linear and anisotropic elastic soft tissue models for medical simulation," in *ICRA2001: IEEE International Conference Robotics and Automation*, (Seoul Korea), May 2001. 6 pages.

[9] S. De, Y.-J. Lim, M. Manivannan, and M. A. Srinivasan, "Physically realistic virtual surgery using the point-associated finite field (PAFF) approach," *Presence: Teleoper. Virtual Environ.*, vol. 15, no. 3, pp. 294–308, 2006.

[10] Y. Zhuang and J. Canny, "Real-time global deformations," in *The fourth International Workshop on Algorithmic Foundations of Robotics (WAFR)*, pp. 97–107, A. K. Peters, 2000.

[11] X. Wu, M. S. Downes, T. Goktekin, and F. Tendick, "Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes," in *EG 2001 Proceedings* (A. Chalmers and T.-M. Rhyne, eds.), vol. 20(3), pp. 349–358, Blackwell Publishing, 2001.

[12] M. Sedef, E. Samur, and C. Basdogan, "Real-time finite-element simulation of linear viscoelastic tissue behavior based on experimental data," *IEEE Comput. Graph. Appl.*, vol. 26, no. 6, pp. 58–68, 2006.

[13] J. Barbič and D. L. James, "Real-time subspace integration for st. venant-kirchhoff deformable models," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, (New York, NY, USA), pp. 982–990, ACM, 2005.

[14] J. Barbič and D. L. James, "Six-dof haptic rendering of contact between geometrically complex reduced deformable models," *EEE Trans. Haptics*, vol. 1, no. 1, pp. 39–52, 2008.

[15] S. Misra, A. M. Okamura, and K. T. Ramesh, "Force feedback is noticeably different for linear versus nonlinear elastic tissue models," in *WHC '07: Proceedings of the Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, (Washington, DC, USA), pp. 519–524, IEEE Computer Society, 2007.

[16] I. Peterlik and A. Krenek, "Haptically driven travelling through conformational space," in *WHC '05: Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Vir tual Environment and Teleoperator Systems*, (Washington, DC, USA), pp. 342–347, IEEE Computer Society, 2005.

[17] E.Samur, M.Sedef, C.Basdogan, L.Avtan, and O.Duzgun, "A robotic indenter for minimally invasive measurement and characterization of soft tissue behavior," *Medical Image Analysis*, vol. 11, no. 4, pp. 361–373, 2007.

[18] I. Amidor, "Scattered data interpolation for electronic imaging systems: a survey," *Journal of Electronic Imaging*, vol. 11, no. 2, pp. 157–176, 2002.

[19] J. Pommier and Y. Renard, "*Getfem++*, an open source generic c++ library for finite element methods." http://www-gmm.insa-toulouse.fr/getfem.

[20] P.R.Amestoy, I.S.Duff, and J.-Y. L'Excellent, "Multifrontal parallel distributed symmetric and unsymmetric solvers," *Comput. Methods in Appl. Mech. Eng.*, vol. 184, pp. 501–520, 2000.

[21] H.Si, *TetGen, A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator*, 2004.